УДК 681.3.001.2

Л.В. ДЕРБУНОВИЧ, д-р техн. наук, проф. НТУ «ХПИ»
И.В. ГОРМАКОВА, аспирант НТУ «ХПИ»

МЕТОДЫ ПОСТРОЕНИЯ АРИФМЕТИЧЕСКИХ МОДУЛЕЙ, ОПЕРИРУЮЩИХ В ПОЛЯХ ГАЛУА

У статті описується новий метод побудови послівно-послідовного помножувача, який базується на поданні елементів поля $GF(2^p)$ у стандартному базисі. Отриманий помножувач має каскадну архітектуру, що легко тестується.. Запропонований помножувач може бути з легкістю побудований для будь-якого поля $GF(2^p)$ та для будь-якого генеруючого полінома F(x).

In this paper a new word-serial multiplier in $GF(2^p)$ for standard-basis representation is developed. Obtained multiplier architecture is scaleable and easy-to-test. Proposed multiplier can be easily designed for any field $GF(2^p)$ and any field-generator polynomial F(x).

Постановка проблемы. В настоящее время потребность в компактных схемах, способных работать с многоразрядными данными, дала толчок к разработке высокоскоростных схем с параллельной обработкой данных. В цифровых устройствах обработки информации, криптосистемах, в системах помехоустойчивого кодирования и т.д. широко используются арифметические модули, функционирующие в полях Галуа $GF\ (2^p)$ [1].

В системах кодирования с обнаружением и исправлением ошибок операции в поле $GF(2^p)$, главным образом суммирование и умножение, являются основными. Например, в [2] показано, что такие операции используются при кодировании и декодировании данных в кодах Рида-Соломона. В [3] показано, что арифметические модули в полях $GF(2^p)$ используется для реализации криптоалгоритмов в эллиптических кривых.

В перечисленнях выше устройствах составными блоками, которые во многом влияют на сложность и время работы системы, являются арифметические блоки. Операция сложения в поле $GF(2^p)$ эквивалентна простой побитовой операции XOR. Однако операция умножения требует более сложной схемной реализации.

Анализ литературы. В [4] были предложены архитектуры сумматоров и умножителей элементов поля, представленных в стандартном и нормальном базисе.

В случае представления элементов поля в нормальном базисе, сумматор может быть построен на основе двух сдвиговых регистров и m XOR вентилей. Умножитель имеет более сложную структуру и состоит из двух сдвиговых регистров и логической схемы, необходимой для вычисления всех компонентов произведения. Достоинством такого метода является применение одной логической схемы для вычисления всех компонентов произведения.

Однако недостатком является использование сдвиговых регистров, имеющих обратные связи.

В [5] представлена архитектура параллельного умножителя в поле $GF(2^p)$ для стандартного базиса представления элементов поля. В параллельных умножителях операция умножения выполняется за один такт, однако аппаратные затраты и площадь на кристалле достаточно велики. В [6, 7] представлены архитектуры пословно-последовательных умножителей. Показано, что такие умножители наилучшим образом соответствует требованиям временных (время выполнения алгоритма умножения), аппаратных (количество логических вентилей) и пространственных (площадь, занимаемая на кристалле) затрат.

Целью статьи является разработка метода синтеза пословно-последовательного умножителя, оперирующего в поле Галуа $GF(2^p)$ и соответствующего требованиям быстродействия, каскадности и тестопригодности.

В предлагаемой архитектуре умножителя выполняется операция умножения по модулю неприводимого полинома, используя так называемый стандартный базис представления элементов поля $GF(2^p)$.

Конечное поле $GF(2^p)$, где p – целое и больше единицы, это числовая система, состоящая из 2^p элементов, в которой правила сложения и умножения соответствуют арифметики по модулю неприводимого полинома степени p с коэффициентами в поле $GF(2)$. Удобство такого поля в практическом применении состоит в том, что каждый элемент поля может быть представлен p двоичными разрядами.

В поле $GF(2^p)$ всегда существует элемент a, который образует все ненулевые элементы поля $\{a, a^2, \ldots, a^{2^p-1}\}$. Элемент $a \in GF(2^p)$ называется образующим элементов поля и является корнем неприводимого полинома $F(x)$, удовлетворяющим условию $F(a)=0$. Неприводимый полином $F(x)$ называют образующим полиномом поля:

$$F(x)= x^p+f_{p-1}x^{p-1}+f_{p-2}x^{p-2}+\ldots+f_1x+1, \quad f_i \in GF(2) \qquad (1)$$

Тот факт, что полином $F(x)$ является неприводимым полиномом, гарантирует, что p элементов $a^0=1, a, a^2, \ldots, a^{p-1}$ поля $GF(2^p)$ линейно независимы в поле $GF(2)$. Таким образом, элементы поля $\{a^0, a, a^2, \ldots, a^{p-1}\}$ образуют стандартный базис представления $\underline{s}$. Произвольный элемент поля B, заданный как двоичный вектор длиной p $[b_0, b_1, b_2, \ldots, b_{p-1}]$, может быть представлен в стандартном базисе как:

$$B=b_0+b_1a+b_2a^2+\ldots+b_{p-1}a^{p-1}=\underline{s}\times[b]^t \qquad (2)$$

где t обозначает операцию транспонирования матрицы двоичных разрядов $[b_0, b_1, b_2, \ldots, b_{p-1}]$.

Пусть заданы два элемента поля A и B, представленные в стандартном базисе следующим образом:

$$A=a_0+a_1a+a_2a^2+\ldots+a_{p-1}a^{p-1}=\sum_{i=0}^{p-1}a_ia^i,\ a_i\in GF(2) \quad (3)$$

$$B=b_0+b_1a+b_2a^2+\ldots+b_{p-1}a^{p-1}=\sum_{i=0}^{p-1}b_ia^i,\ b_i\in GF(2) \quad (4)$$

Тогда произведение C элементов поля A и B может быть задано как

$$C=AB \bmod F(\alpha)=(\sum_{i=0}^{p-1}a_ia^iB)\bmod F(\alpha) \quad (5)$$

При пословно-последовательном алгоритме умножении элементов поля один из операндов разбивается на слова. Разделим операнд A на $\lceil p/\omega \rceil=k$ слов длиной в ω бит. Тогда операнд A может быть представлен в виде следующего полинома:

$$A=A_{k-1}a^{(k-1)\omega}+\ldots+A_2a^{2\omega}+A_1a^{\omega}+A_0 \quad (6)$$

где A_j – полином степени $\leq(\omega-1)$, $j=0,\ldots,(k-1)$. Причем степень полинома A_{k-1} может быть меньше, чем $(\omega-1)$.

Каждое полученное слово в свою очередь также может быть представлено в виде полинома:

$$A_j=a_{\omega j+(\omega-1)}a^{(\omega-1)}+a_{\omega j+(\omega-2)}a^{(\omega-2)}+\ldots+a_{\omega j+2}a^2+a_{\omega j+1}a+a_{\omega j},\ j=0,\ldots,(k-1) \quad (7)$$

Тогда произведение C элементов поля A и B можно записать как:

$$C=[(\mathbf{K}(A_{k-1}Ba^{w}+A_{k-2}B)a^{w}+\mathbf{K}+A_1B)a^{w}+A_0B]\bmod F(a) \quad (8)$$

Сформулируем алгоритм синтеза пословно-последовательного умножителя в поле $GF(2^p)$.

Входные данные: элементы поля $A,\ B\in GF(2^p)$, образующий полином $F(\alpha)$

Выходные данные: $C=AB \bmod F(\alpha)$

ШАГ 1: Установить $C_{-1}=0$

ШАГ 2: Установить счетчик $i=0$. Для $[i=0\div(k-1)]$ повторить следующую последовательность действий:

ШАГ 3: Вычислить значение $D_i=A_{(k-1)-i}B \bmod F(\alpha)$

ШАГ 4: Вычислить значение $C_i=C_{i-1}\cdot a^{\omega}+D_i$

ШАГ 5: Увеличить значение счетчика i на 1. Если $i<k$, перейти к шагу 3, иначе перейти к шагу 6.

ШАГ 6: Присвоить $C=C_{(k-1)}$

ШАГ 7: Конец алгоритма.

На шаге 3-4 для каждого слова $A_{(k-1)-i}$, $i=0\ldots(k-1)$ выполняется вычисление частичного произведения. На шаге 3 выполняется умножение по модулю $F(\alpha)$ текущего слова A_j на второй операнд B:

$$D_i=A_jB=a_0B+a_1Ba+a_2Ba^2+\ldots+a_{\omega-1}Ba^{\omega-1} \quad (9)$$

На шаге 4 выполняется умножение вычисленной на предыдущем шаге итерации суммы C_{i-1} на $\cdot a^{\omega}$, затем полученный результат складывается с произведением A_jB, вычисленным на шаге 3. Полученный результат присваивается переменной C_i.

На основании приведенного метода была разработана структурная схема пословно-последовательного умножителя (рис.1).

В состав пословно-последовательного умножителя входят следующие блоки:

1) СКА – сети клеточных автоматов;
2) сдвиговый регистр с первым операндом A;
3) AND сеть;
4) XOR сеть;
5) блок суммирования промежуточных результатов D_i и C_i;
6) блок вычисления произведения $Y=L\cdot a^{\omega} \bmod F(\alpha)$;
7) регистр R.

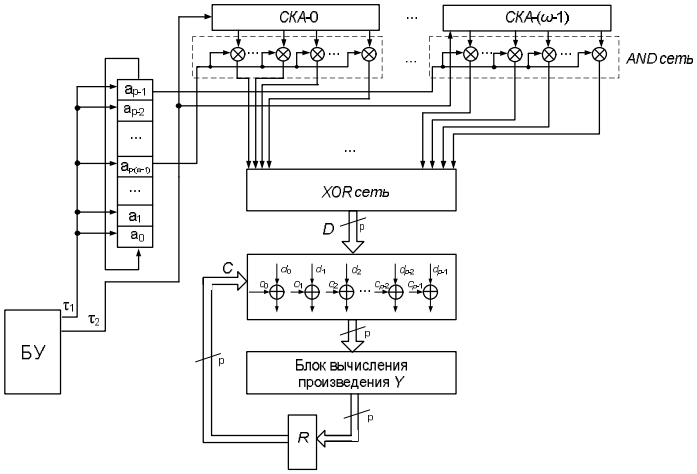

Рис.1 Структурная схема пословно-последовательного умножителя

СКА предназначены для последовательного вычисления произведений B, Ba, …, $Ba^{\omega-1}$ за w тактов. После такта w функционирование СКА прекращается. Вычисленные значения B, Ba, …, $Ba^{\omega-1}$ хранятся соответственно в СКА-0, СКА-1, …, СКА-(ω-1). Каждая СКА представляет собой однородную сеть из p ячеек. Выход (p-1)-ой ячейки заводится на вход нулевой ячейки. Остальные ячейки $z_1,\ldots, z_{p-1}$ имеют одинаковую структуру: каждая ячейка связана с соседом слева z^1, кроме того, в ячейке добавлен верхний вход z^2. На верхний вход z^2 i-ой ячейки поступает выходной сигнал (p-1)-ой ячейки только в том случае, если коэффициент f_i образующего полинома $F(x)$ равен 1.

Каждая из w AND сетей состоит из p двухвходовых вентилей AND. Один из входов i-го AND вентиля m-ой AND сети запитан выходом i-ой ячейки m-ой СКА. На второй вход всех вентилей m-ой AND сети по общей одноразрядной шине подается один бит a_m из входного слова A_j.

XOR сеть предназначена для последовательного суммирования произведений a_0B, a_1Ba, a_2Ba^2,…, $a_{\omega-1}Ba^{\omega-1}$. На выходе XOR сети формируется частичное произведение $D=A_jB \bmod F(\alpha)$.

Блок суммирования промежуточных результатов состоит из p двухвходовых вентилей XOR. Этот блок предназначен для побитового суммирования двух операндов $L=D \oplus C$: первый операнд D – частичное произведение, второй операнд C – содержимое регистра R.

Блок вычисления произведения Y представляет собой специальным образом построенную сеть из XOR вентилей, которая обеспечивает вычисление произведения $Y=L \cdot a^{\omega} \bmod F(\alpha)$.

Работу умножителя можно описать следующим образом. На нулевом такте в СКА-0, СКА-1, …, СКА-(ω-1) загружается операнд B, регистр R обнуляется. На первом такте на выходах всех СКА будет значение B. Функционирование СКА-0 после первого такта прекращается. Таким образом, состояние СКА-0 далее остается неизменным и равно B. На втором такте на выходах СКА-1, …, СКА-(ω-1) будет значение Ba и прекращается функционирование СКА-1. На такте ω на выходах СКА-0, СКА-1, …, СКА-(ω-1) будет соответственно значения B, Ba, …, $Ba^{\omega-1}$, которые сохраняются до окончания выполнения умножения двух элементов поля. После каждого такта выходные значения СКА поступают на входы AND сетей, которые последовательно вычисляют произведения a_iBa^i для текущего слова A_{k-1}. Выходы AND сетей заводятся на XOR сеть.

На такте ω в XOR сети происходит суммирование всех вычисленных произведений a_0B, a_1Ba, a_2Ba^2,…, $a_{\omega-1}Ba^{\omega-1}$. Далее вычисленное значение D поступает на блок суммирования промежуточных результатов, в котором происходит суммирование с содержимым регистра R. Так как содержимое регистра равно нулю, то на выходе блока суммирования промежуточных результатов получаем значение $L=A_{k-1}B$. Далее значение L поступает на вход блока вычисления произведения $Y=L \cdot a^{\omega} \bmod F(\alpha)$. Выходным значением блока на такте ω будет значение $Y=A_{k-1}B \cdot a^{\omega} \bmod F(\alpha)$, которое записывается в регистр R.

На такте (ω+1) происходит циклический сдвиг регистра с первым операндом A влево на ω разрядов. Следовательно, на входы AND сетей поступают биты слова A_{k-2}. Далее для слова A_{k-2} выполняется та же последовательность действий, что и для слова A_{k-1}. Таким образом, на выходе блока суммирования промежуточных результатов получаем значение $L=A_{k-1}B \cdot a^{\omega} \bmod F(\alpha)+A_{k-2}B$. Выходным значением блока вычисления произведения на такте (ω+1) будет значение $Y=(A_{k-1}B \cdot a^{\omega} \bmod F(\alpha)+A_{k-2}B) \cdot a^{\omega} \bmod F(\alpha)$, которое записывается в регистр R. На протяжении последующих тактов выполняются аналогичные операции для последующих слов A_{k-3}, …, $A_{k-\omega}$. Таким образом, на такте 2ω в регистр R будет записан результат вычисленного выражения:

$$C = [(\mathbf{K}(A_{k-1}Ba^{w} + A_{k-2}B)a^{w} + \mathbf{K} + A_{k-w}B)a^{w} \bmod F(a) \qquad (10)$$

Общее время работы умножителя составляет ($\lceil p/\omega \rceil + \omega$) тактов. На последнем такте с выхода блока суммирования снимается значение произведения, соответствующее формуле (8).

Представленная архитектура пословно-последовательного умножителя в поле $GF(2^p)$ включает в себя унифицированные блоки и позволяет просто реализовать умножитель на ПЛИС типа FPGA, что дает возможность легко модифицировать архитектуру умножителя при изменении длины операндов, длины слова, образующего полинома поля. Изменение образующего полинома при сохранении степени полинома p требует лишь изменения правил настройки сети клеточных автоматов, входящих в состав умножителя, при полном сохранении их структуры.

В настоящее время разрабатываются новые методы построения умножителей. Основное требование к разрабатываемым архитектурам арифметических блоков – снижение количества вентилей и площади схемы умножителей.

Важно также обратить внимание на вычислительную сложность алгоритма для оценки максимального значения p, с которым можно работать при данном методе. Этот аспект важен в криптографических устройствах, в которых p лежит в диапазоне 160÷521 [5].

Выводы. Предложенный метод синтеза пословно-последовательного умножителя, оперирующего в поле Галуа $GF(2^p)$, позволяет синтезировать архитектуру умножителя, которая соответствует требованиям быстродействия, каскадности и тестопригодности.

Список литературы: 1. *A. J. Menezes, I. F. Blake, X. Gao, R. C. Mullin, S. A. Vanstone, and T. Yaghoobian*. Applications of finite fields. Boston: Kluwer Academic, 1993. 2. *D.V. Sarwate and N.R. Shanbhag*. High-speed architecture for Reed-Solomon decoder. IEEE Trans. Very Large Scale Integration (VLSI) Systems, vol. 9, no. 7, pp.641-655, oct. 2001. 3. *Souichi Okada, Naoya Torii, Kouichi Itoh, Masahiko Takenaka*. Implementation of Elliptic Curve Cryptographic Coprocessor over $GF(2^m)$ on an FPGA. C.K.Kocand C.Paar(Eds.): CHES 2000, LNCS 1965, pp. 25-40, 2000. Springer-VerlagBerlin Heidelberg 2000. 4. *J. Omura and J. Massey*. Computational method and apparatus for finite field arithmetic. US Patent Number 4, 587, 627, May 1986.. 5. *N. Petra, D. De Caro and A.G.M. Strollo*. A novel architecture for Galois Fields $GF(2^m)$ multipliers based on Mastrovito scheme. IEEE Trans.Comput., 2007, Nov., vol. 56, pp.1470-1483. 6. *G. Orlando, C. Paar*. A High Performance Reconfigurable Elliptic Curve Processor for $GF(2^m)$. Proc. Second Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '00), K. Koc and C. Paar, eds., pp. 41-56, 2000. 7. *Hans Eberle, Sheueling Chang, Nils Gura, Sumit Gupta, Dniel Finchelstein, Edouard Goupy, Douglas Stebila*. An End-to-End Systems Approach to Elliptic Curve Cryptography. Sun Microsytems Laboratories 2002-2003.

Поступила в редакцию 20.04.10